Tracing Crease Curves by Solving a System of Differential Equations

Antonio M. López, Joan Serrat

Computer Vision Center
Departamento de Informática, Edificio C
Universidad Autónoma de Barcelona,
08193 Bellaterra, Spain
tel: +34 3 5811863, Fax: +34 3 5811670
e-mail : {antonio | joans}@upisun1.uab.es

Abstract. Different kinds of digital images can be modelled as the sampling of a continuous surface, being described and analyzed through the extraction of geometric features from the underlying surface. Among them, ridges and valleys or, generically, creases, have deserved special interest. The computer vision community has been relying on different crease definitions, some of them equivalent. Although they are quite valuable in a number of applications, they usually do not correspond to the real creases of a topographic relief. These definitions give rise either to algorithms that label pixels as crease points, and then focus on the problem of grouping them into curves, or to operators whose outcome is a creaseness image. We draw our attention to the real crease definition for a landscape, due to Rudolf Rothe, which is based on the convergence of slopelines. They are computed by numerically solving a system of differential equations. Afterwards, we extract Rothe creases which are parts of slopelines where others converge, avoiding in such a way any pixel-grouping step. At the same time we compute a creaseness image according to this definition.

Key Words : ridge, valley, creaseness, slopelines, differential equation.

1 Introduction

In computer vision there is an image model underlying every image analysis method. One of these models considers images as the sampling of a manifold, such as a graphic surface in the case of two–dimensional images. Due to the fact that dealing with a surface as a whole is, in general, computationally infeasible, one must look for manifolds of lesser dimension confined in it having a visual meaning or serving as image descriptors. Among them, ridges and valleys or, generically, *creases*, have been widely used for several purposes. For instance, they are involved in the description and segmentation of medical images [3, 5], and digital elevation models [7, 14]. They have also been shown to be useful for 2D and 3D medical image registration [2, 13], since they correspond to anatomic features. Creases are also especially useful for range image processing [4, 11], because these images are true sampled surfaces.

In computer vision literature, we find diverse crease characterisations, some of them equivalent. In [1] we find a rather complete classification according to several criteria. In essence, we distinguish three main classes : implementations of definitions from differential geometry that 1) take into account or 2) not, the existence of a singled out direction, and 3) algorithmic or constructive definitions.

Even if all these definitions and their implementations are very useful in a number of applications, they share one or several of the following drawbacks :

1. None of them are the true creases in the sense of ridges and valleys of a topographic relief; that is, the loci of points where water gathers to run downhill, in the case of valleys, and similarly for ridges but with the same relief turned upside down.
2. Implementations usually do not compute curves, that is, sequences of adjacent pixel coordinates, but images where pixels are labelled as ridge, valley or background. Therefore, a further error prone process of grouping is necessary.
3. Crease operators give relatively high responses in the neighbourhood of a crease, so an additional decision rule (e.g. thresholding) must be applied, which commonly produces thick lines. These operators also tend to have high outputs at regions which exhibit high curvature but are not necessarily creases.
4. Some implementations rely on critical points like maxima, minima or saddle points which are quite dependent on noise. As a consequence, creases are unstable.

In this paper, we present a new method for crease extraction from two–dimensional images that overcomes all of the former problems. Moreover, this method is based on the true characterisation of creases in the topographic sense, due to Rudolf Rothe [12], which is discussed in detail in [8]. In order to compute Rothe creases, we first realize that curves lying on a surface can be formulated as the solution of a system of coupled Ordinary Differential Equations (ODEs). It is solvable by a standard numerical integration 'engine', which only needs to be 'fuelled' with values depending on surface derivatives. Some useful curves in computer vision such as parabolic curves, curves of mean curvature extrema, the classical ridge and valley curves and many others [9], can be expressed and computed in this way. Here, we shall focus on *slopelines*, because they converge towards Rothe creases. Therefore, we have devised a convergence assessment procedure for discrete curves.

This article is organized as follows. Section 2 reviews different crease definitions. In Sect. 3 we formulate curves on a parametric surface in $\mathbb{R}^3$ as the solution of a system of coupled ODEs. Next, in Sect. 4, we present an algorithm to extract Rothe creases from slopelines. Section 5 illustrates the results obtained. Finally, in Sect. 6 we discuss the conclusions and future work.

2 Different Crease Definitions

2.1 Definitions Independent of a Singled Out Direction

They are based on the principal curvatures of the surface which are independent of how the surface is embedded in the space. For instance, a crest line (ridge or valley) of a surface given in implicit form is defined in [13] as the extrema of the maximum principal curvature in absolute value $k_{\max}$, in its principal direction $\mathbf{v}_{\max}$, that is, $\nabla k_{\max} \cdot \mathbf{v}_{\max} = 0$ where $\nabla k_{\max}$ is the gradient of $k_{\max}$. Ridges are distinguished from valleys by the sign of the mean curvature k_{M},which, conversely to the principal curvatures, is a non–directional quantity. It provides information about the concavity ($k_{\mathrm{M}} > 0$, valley) or convexity ($k_{\mathrm{M}} < 0$, ridge) of the surface. Actually, creases have been alternatively defined as the extrema of the mean curvature [1, 11].

These definitions are suitable for spaces where there is not a privileged direction. They have the advantage of rotational invariance because they are based on principal curvatures. Hence, they have also been used in the context of spaces with a privileged direction, e.g. for object recognition in range images [4, 11].

2.2 Creases Based on a Singled Out Direction

In this case we model images as a height function, being the height axis the singled out direction. In 2D images we have a function $I(x_1, x_2)$ where the singled out direction is most often the intensity axis. Now, we distinguish the crease characterisation due to De Saint–Venant from the one due to Rothe.

The De Saint–Venant condition. Several authors [2, 6] have taken *crease* points of a function $I(x_1, x_2)$ as the height extrema in the directions where their second directional derivative is also extreme. Creases have also been identified as the loci of curvature extrema of the level curves (isohypses, isophotes). This definition has given rise to several implementations [3, 7] and the same idea has been extended to three–dimensional implicit surfaces [13].

Actually, both characterisations are equivalent and correspond to the condition of creases given by De Saint–Venant [8] as the loci of extreme slope along a level curve. If we denote the first and second order partial derivatives of $I(x_1, x_2)$ by $I_{x_\alpha} = \partial I(x_1, x_2)/\partial x_\alpha$ and $I_{x_\alpha x_\beta} = \partial^2 I(x_1, x_2)/\partial x_\alpha \partial x_\beta$, respectively, the gradient of $I(x_1, x_2)$ will be $\nabla I = (I_{x_1}, I_{x_2})$, the vector orthogonal to it $\nabla_\perp I = (I_{x_2}, -I_{x_1})$ and the Hessian of $I(x_1, x_2)$

$$\nabla\nabla I = \begin{bmatrix} I_{x_1x_1} & I_{x_1x_2} \\ I_{x_1x_2} & I_{x_2x_2} \end{bmatrix} . \tag{1}$$

Then, taking the magnitude of the gradient as a slope measure, the De Saint–Venant condition is expressed as

$$\frac{\nabla I \cdot \nabla\nabla I \cdot \nabla_\perp I^{\mathrm{T}}}{\| \nabla I \|^2} = 0 . \tag{2}$$

Rothe's characterisation. Koenderink and van Doorn saved from oblivion the fact that the former definition does not correspond to the intuitive notion of creases as steepest descent/ascent water paths in a landscape [8]. Breton de Champ proved that the only curves satisfying both conditions are confined to vertical planes. However, it is clear that in general this is not true. We owe to R. Rothe the right ridge and valley characterisation as parts of slopelines where other slopelines converge. These parts are also referred as special slopelines. The family of slopeline curves, also called creeplines or flowlines, is defined as being orthogonal to the family of level curves. That is to say, those following the gradient direction. We shall see in Sect. 4 that solving the slopeline differential equation

$$\nabla_{\perp} I \cdot \mathrm{d}\mathbf{x} = 0 \tag{3}$$

for initial points spread all over the image, the computed curves converge to all the salient creases.

2.3 Algorithmic Definitions

They are based on critical points of the image seen as a landscape, namely, local maxima, minima and saddle points. Therefore, they also assume the existence of a privileged direction. Perhaps, the most well known algorithmic definition is the morphological watershed which computes regions of influence of local minima [14]. Each region is claimed to be a catch basin and their closed borders are identified as ridges or divide lines in the sense of a topographic relief. Another algorithmic definition consists in tracing curves that join critical points following the gradient direction [5]. These algorithms suffer from the instability of critical points, that may easily appear, disappear or move due to noise. In addition, they also fail to match the true creases, despite its undeniable usefulness.

In all these characterisations, crease points are classified as ridges or valleys, depending on the sign of the second directional derivative of the height function $I(x_1, x_2)$ in the direction orthogonal to the gradient :

$$I_{\nabla_{\perp}\nabla_{\perp}} = \frac{\nabla_{\perp} I \cdot \nabla\nabla I \cdot \nabla_{\perp} I^{\mathrm{T}}}{\| \nabla I \|^2} \ . \tag{4}$$

$I_{\nabla_{\perp}\nabla_{\perp}}$ is negative at convex regions (ridges) and positive at concave regions (valleys). As a matter of fact, there are convolution operators which approximate $I_{\nabla_{\perp}\nabla_{\perp}}$ at a given scale in such a way that points with an output magnitude above a certain threshold are considered creases [2], implementing in this way an approximation to the De Saint-Venant condition.

3 Curves on Surfaces as Coupled ODEs

Let us consider a parametric surface $\mathbf{s}(x_1, x_2)$ in $\mathbb{R}^3$, and a curve $\mathbf{s}(\mathbf{x}(t))$ lying on it, for $\mathbf{x}(t) = (x_1(t), x_2(t))$ on the plane of parametrisation $x_1 x_2$. $\mathbf{x}(t)$ is

completely determined by a certain relationship between parameters x_1 and x_2. This relationship may take the form of an ODE

$$\mathbf{f} \cdot \mathrm{d}\mathbf{x} = 0 \tag{5}$$

where $\mathbf{f} = (f_1(x_1, x_2), f_2(x_1, x_2))$, for certain functions f_1, f_2, and $\mathrm{d}\mathbf{x} = (\mathrm{d}x_1, \mathrm{d}x_2)$. This is the case of (3), the slopeline ODE, where $f_1 = I_{x_2}$ and $f_2 = -I_{x_1}$.

To solve this equation means to find the integral curves of the vectorial field $\mathbf{w} = (f_2(x_1, x_2), -f_1(x_1, x_2))$, on the plane of parametrisation, orthogonal to the vectorial field $\mathbf{f}$. That is to say, to find curves $\mathbf{x}(t)$ which are the solution of the following system of coupled ODEs

$$\mathrm{d}\mathbf{x}(t)/\mathrm{d}t = (\mathrm{d}x_1(t)/\mathrm{d}t, \mathrm{d}x_2(t)/\mathrm{d}t) = \mathbf{w} \tag{6}$$

for *some unknown* parametrisation t. That is, the t that matches at each point $(x_1(t), x_2(t))$ the tangent of the curve with the direction, *and* the magnitude of $\mathbf{w}$. However, we know that two possible parametrisations are the arclength s of the curve $\mathbf{x}(t)$ lying on the x_1x_2–plane and of the curve $\mathbf{s}(\mathbf{x}(t))$ lying on the surface. Then, we can write

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = \pm \frac{\mathbf{w}}{\sqrt{\mathbf{w} \cdot \mathbf{G} \cdot \mathbf{w}^{\mathrm{T}}}} \tag{7}$$

where $\mathbf{G}$ is in the first case the identity matrix and in the second case the matrix with the covariant components of the surface metric tensor, this is,

$$\mathbf{G} = \begin{bmatrix} 1 + I_{x_1} & I_{x_1} I_{x_2} \\ I_{x_1} I_{x_2} & 1 + I_{x_2} \end{bmatrix} . \tag{8}$$

We can solve (7) with a numerical integration method. In particular, we use the fourth order Runge–Kutta with adaptative step [10]. In order to start the numerical integration and obtain curve points, only an initial point and the component functions of $\mathbf{w}$, are needed.

In this paper, we are interested in curves on the plane of parametrisation because in two–dimensional images they are the projection of the corresponding curves on the surface. Hence, we should parametrise by the arclength of $\mathbf{x}(t)$, which has the advantage over the parametrisation by $\mathbf{s}(\mathbf{x}(t))$ that the solution curve runs faster because $\mathbf{w} \cdot \mathbf{w} \leq \mathbf{w} \cdot \mathbf{G} \cdot \mathbf{w}^{\mathrm{T}}$. However, the integration method has a local error of order $\mathcal{O}(\Delta s^5)$ for an increment Δs of the parameter variable. Thus, a higher speed implies a greater local error in each step. This in turn may involve a computation overhead because the integration process can be compelled to try too many fractions of Δs in order not to exceed the allowed local error. Therefore, we have chosen the second parametrisation.

4 Extraction of Rothe Creases

We have seen that the characterisation of creases in a continuous landscape identifies them as (parts of) slopelines where other slopelines converge. In the discrete case, we consider that creases are formed by contiguous slopeline segments, where there is a high degree of convergence. More precisely, we shall consider that a set of sampled slopelines converge into a certain slopeline segment if all of them overlap in it. The more curves crowd together, the higher the convergence is. The central idea of the algorithm is to compute a large number of slopelines and store them as separate curves. Then for each one, the segments with high convergence are selected. Finally, redundant segments similar to and shorter than other selected segments are discarded. The algorithm has the following steps: 1) Slopeline extraction, 2) Formation of the creaseness image, and 3) Crease extraction.

Slopeline extraction. To determine how many slopelines have to be traced and from which starting points, we note that too few curves will cause a poor overlapping and therefore some creases will surely be missed, and that too many curves just slightly improves the result. Hence, we limit the number of computed curves to those which are needed in order to cover the whole image, making sure that at least one curve passes through each pixel. Thus, at any given time, a pixel is liable to be an initial point $\mathbf{x}(0)$ for (7) if no curve has visited it yet. By following this rule, we are able to find the main and wider creases in several kinds of images. However, in images with small details, sometimes narrow creases are missed. We have overcome this problem by working at subpixel resolution. This means that, if it is necessary, slopelines are sampled not at integer coordinates but at a finer resolution r, being for instance $r = 2$ for double resolution. Values of $r = 2, 3$ are sufficient to produce good results in these cases, as we shall see in Sect. 5.

To integrate the ODEs system of the slopelines numerically, the values of the first partial derivatives of the image are required at points with integer coordinates but also at points in between. We approximate the derivative of the image at a certain scale σ, at which creases are extracted, by the convolution with the sampled derivative of a bidimensional Gaussian of variance σ^2. For first order derivatives, $I_{x_\alpha}(x_1, x_2; \sigma) \approx I(x_1, x_2) * G_{x_\alpha}(x_1, x_2; \sigma), \quad \alpha = 1, 2$. Note that this equation applies only to pixel coordinates, that is, both x_1 and x_2 are integers. Elsewhere, we approximate subpixel derivatives evaluating the derivatives of an algebraic polinomial, obtained by a bicubic interpolation [10] from the four nearest pixels.

Formation of the creaseness image. The convergence of curves in the continuous space is interpreted as the overlapping of sampled curves and the discrete ones. Thus, we need to count, for each point of the image domain, how much slopelines pass through it. This process must be done at resolution r, therefore in a matrix r^2 times larger than the original image. This matrix has an interesting

meaning : it is a *creaseness* image, analogously to images produced by crease operators like $I_{\nabla_\perp \nabla_\perp}$ (Fig. 1e), though corresponding to Rothe creases (Fig. 1d).

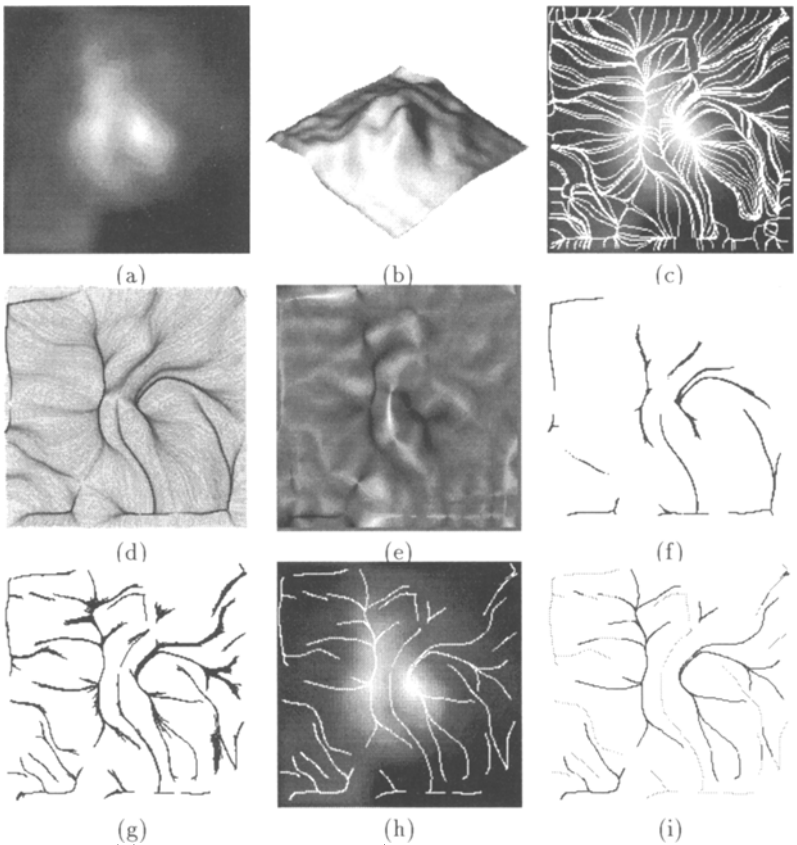


Fig. 1. (a) Heart scintigraphic image. (b) Image seen as a landscape. (c) Several slopelines for $r = 3$ and $\sigma = 1.0$ pixels. (d) Creaseness image where darker grey-levels denote greater accumulation. (e) Operator $I_{\nabla_\perp \nabla_\perp}$ applied to (a), displaying convexity (dark) and concavity (bright). (f) Thresholding of (d) showing regions of greater accumulation. (g) Selected slopeline segments for $L = T = 4$. (h) Special slopelines for $P = 0.95$. (g) Classification into ridges (black) and valleys (grey) according to the sign of $I_{\nabla_\perp \nabla_\perp}$.

Crease extraction. Creases are segments of certain slopelines where there is a high convergence. Hence, we select those segments from the stored slopelines where accumulation at each point is greater than a threshold T. We impose a further condition in order to avoid small segments, most probably due to the sampling of slopelines which do not converge but just get closer than $1/r$ pixels and then run far away : those segments must be longer than rL, for another threshold parameter L.

Finally, if we look at the image of segments passing the two conditions, it displays bunches of segments, not isolated curves (Fig. 1g). The reason is precisely that they group because of their high accumulation one over the others, thus fulfilling the two conditions above. In order to discard redundant segments, we apply the following rule : given two segments, we eliminate the shorter one if it overlaps the longer one in more than a certain length fraction P.

The results are not very sensitive to parameters T, L and P. We mean that they are just minimum values intended to get rid of several abnormal cases like very short or very weak creases. Typical values are $T = 3, L = 3$ and $P = 0.9$. The increase of T and L, or the decrease of P, do not change substantially the final result.

5 Results

We now present the results obtained for Rothe's definition of ridge and valley in a landscape. We shall illustrate its utility in the context of two applications on different medical image modalities: coronary arteriography and brain MRI. In addition, results obtained for two range images are also presented.

In coronary arteriography the goal is to delineate its vessels. We have detected them searching the ridges of the image. In fact, thin vessels *are* ridges. Figure 2b shows the extracted ridges superimposed to the image at the same scale σ for which they were computed.

The second application is the extraction of salient features to be used in the registration of two MR or MR and CT brain images. In Fig. 3b we see the ridges and valleys that have been extracted by our algorithm. They can be used to calculate the geometric transform between a pair of images of the same slice, by means of some process of curve matching. Alternatively, it is possible to obtain the registration transform from the correlation of the creaseness images (Fig. 3c) of the two slices as it is done in [2] with the $I_{\nabla_\perp \nabla_\perp}$ operator (Fig. 3d).

On the other hand, we have experimented with our algorithm in some range images. Figure 4b shows the creases of the simple range image of a block. They perfectly match its roof borders. We have also tested the algorithm with a Digital Elevation Model image (Fig. 4c), which is a real topographic relief.

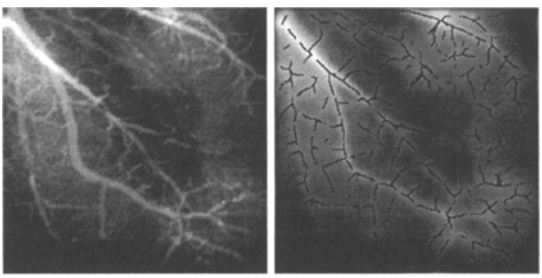
In the previous examples, such as in Fig. 2b, we observe short disconnected segments that could be joined in order to produce longer, perceptually better crease curves. This fragmentation is caused by the saddle points of the image, that, by definition, slopelines can not reach nor cross. Formally, there is nothing wrong with this, but in many applications it should be convenient to link these segments.

The computation times depend mainly on the image size and the resolution r. For instance, in a SPARC 10 computer, the creases of Fig. 1a which is 64×64 pixels, have been computed in 1 minute at a resolution of $r = 3$ whereas for $r = 1$ the time is 15 seconds.

6 Conclusions and Future Work

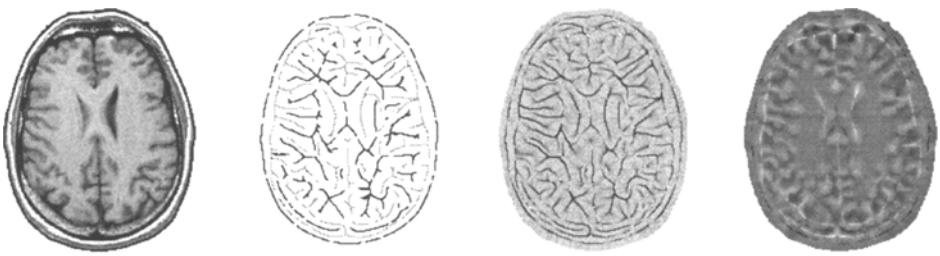
In this paper we proposed a new method to extract ridges and valleys or, generically, creases, of a landscape according to the correct definition by Rothe. We have developed an algorithm which obtains these creases, first covering the whole image with slopelines and then extracting from them the segments where they converge. In this way, we obtain creases not as a binary image but as a set of curves, avoiding a further grouping step. At the same time our algorithm extracts a creaseness image, according to this definition.

As a future work we will join Rothe creases through saddle points, because it is interesting for most applications. Finally, we will study the evolution of Rothe creases in scale space.



(a) (b)

Fig. 2. (a) Coronary arteriography. (b) Ridges for $\sigma = 3.0$ pixels, $r = 1$



(a) (b) (c) (d)

Fig. 3. (a) Brain MRI (axial slice). (b) Ridges (black) and valleys (grey) for $\sigma = 3.0$ pixels, $r = 1$ (c) Creaseness image from Rothe's definition. (d) Response to the $I_{\nabla_\perp \nabla_\perp}$ operator at the same scale.

References

1. D. Eberly, R. Gardner, B. Morse, S. Pizer, C. Scharlach. *Ridges for Image Analysis. Journal of Mathematical Imaging and Vision*, **4**, N. 4, p. 353–373, 1994
2. P. A. Van den Elsen, J. B. A. Maintz, E. D. Pol, M. A. Viergever. *Automatic Registration of CT and MR Brain Images Using Correlation of Geometrical Features. IEEE Trans. on Medical Imaging*, **14**, N. 2, p. 384–396, June 1995.

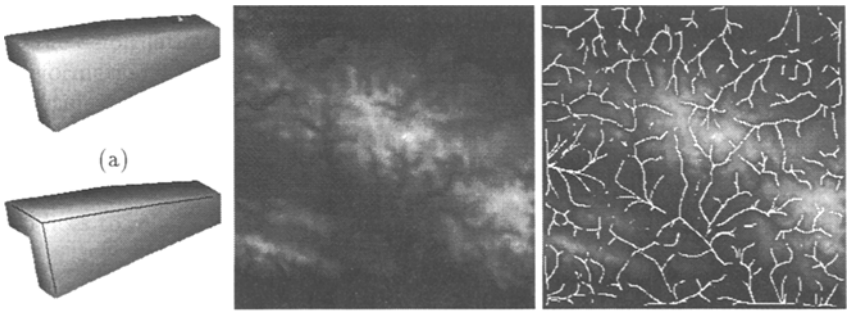


Fig. 4. (a) Range image of a block. (b) Rothe ridges for $\sigma = 1.5$ pixels, $r = 1$. (c) Digital Elevation Model fragment of a central region of Haiti. (d) Valleys for $\sigma = 1.0$ pixels, $r = 2$.

3. J. M. Gauch, S. M. Pizer. *Multiresolution analysis of ridges and valleys in grey-scale image. IEEE Trans. on Pattern Analysis and Machine Intelligence* , **15**, N. 6, p. 635–646, June 1993.
4. G. G. Gordon. *Face Recognition based on Maps and Surface Curvature.* In SPIE **1570** Geometric Methods in Computer Vision, p. 234–247, 1991.
5. L. D. Griffin, A. C. F. Colchester, G. P. Robinson. *Scale and Segmentation of Grey-Level Images using Maximum Gradient Paths.* In Information Processing in Medical Imaging (IPMI), p. 256–272, July 1991. Lecture Notes in Computer Science, **511**, Springer-Verlag.
6. R. M. Haralik, L. G. Shapiro. *Computer and Robot Vision* (**1**). Addison-Wesley, 1992.
7. I. S. Kweon, T. Kanade. *Extracting Topographic Terrain Features from Elevation Maps. CVGIP–Image Understanding*, **59**, N. 2, p. 171–182, March 1994.
8. J. J. Koenderink, A. J. van Doorn. *Local Features of smooth Shapes: Ridges and Courses.* In SPIE **2031** Geometric Methods in Computer Vision II, p. 2–13, 1993.
9. A. M. Lopez, J. Serrat. *Image Analysis Through Surface Geometric Descriptors.* In Proceedings of the VI Spanish Symposium on Pattern Recognition and Image Analysis, p. 35–42, April 1995.
10. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C*, 2nd Edition Cambridge University Press, 1992.
11. S. Panikanti, C. Dorai, A. K. Jain. *Robust feature detection for 3D object recognition and matching.* In SPIE **2031** Geometric Methods in Compuetr Vision, p. 366–377, 1993.
12. R. Rothe. *Zum Problem des Talwegs. Sitzungsber. Berliner Math. Gesellschaft*, **14**, p. 51–69, 1915.
13. J. P. Thirion, A. Gourdon. *Computing the Differential Characteristics of Isointensity Surfaces. CVGIP–Image Understanding*, **61**, N. 2, p. 190–202, March 1995.
14. L. Vincent, P. Soille. *Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. IEEE Trans. on Pattern Analysis and Machine Intelligence* , **13**, N. 6, p. 583–598, June 1991.